Empirical Evaluation of Approximation Algorithms for Generalized Graph Coloring and Uniform Quasi-Wideness

Wojciech Nadara[1]
Institute of Informatics, University of Warsaw, Poland
wn341489@students.mimuw.edu.pl

Marcin Pilipczuk[2]
Institute of Informatics, University of Warsaw, Poland
malcin@mimuw.edu.pl

Roman Rabinovich[3]
Lehrstuhl für Logic und Semantik, Technische Universität Berlin, Germany
roman.rabinovich@tu-berlin.de

Felix Reidl
Department of Computer Science, Royal Holloway University of London, UK
felix.reidl@rhul.ac.uk

Sebastian Siebertz[4]
Institute of Informatics, University of Warsaw, Poland
siebertz@mimuw.edu.pl

Abstract

The notions of *bounded expansion* and *nowhere denseness* not only offer robust and general definitions of uniform sparseness of graphs, they also describe the tractability boundary for several important algorithmic questions. In this paper we study two structural properties of these graph classes that are of particular importance in this context, namely the property of having bounded *generalized coloring numbers* and the property of being *uniformly quasi-wide*. We provide experimental evaluations of several algorithms that approximate these parameters on real-world graphs. On the theoretical side, we provide a new algorithm for uniform quasi-wideness with polynomial size guarantees in graph classes of bounded expansion and show a lower bound indicating that the guarantees of this algorithm are close to optimal in graph classes with fixed excluded minor.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases Empirical Evaluation of Algorithms, Sparse Graph Classes, Generalized Coloring Numbers, Uniform Quasi-Wideness

[1] Supported by the "Recent trends in kernelization: theory and experimental evaluation" project, carried out within the Homing programme of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund.

[2] Supported by the "Recent trends in kernelization: theory and experimental evaluation" project, carried out within the Homing programme of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund.

[3] Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC Consolidator Grant DISTRUCT, grant agreement No 648527).

[4] Supported by the National Science Centre of Poland via POLONEZ grant agreement UMO-2015/19/P/ST6/03998, which has received funding from the European Union's Horizon 2020 research and innovation programme (Marie Skłodowska-Curie grant agreement No. 665778).

Digital Object Identifier 10.4230/LIPIcs.SEA.2018.14

Related Version A full version of the paper is available at `https://arxiv.org/abs/1802.09801`.

Acknowledgements We thank Christoph Dittmann for providing us with his code for the `mfcs` algorithm, which we partially used for our implementation. We thank Michał Pilipczuk for many hours of fruitful discussions.

1 Introduction

The exploitation of structural properties found in sparse graphs has a long and fruitful history in the design of efficient algorithms. Besides the long list of results on planar graphs and graphs of bounded degree (which are too numerous to fairly represent here), the celebrated structure theory of graphs with excluded minors, developed by Robertson and Seymour [57] falls into this category. It not only had an immense influence on the design of efficient algorithms (see e.g. [18, 19]) it further introduced the now widely used notion of treewidth (see e.g. [8]) and gave rise to the field of parameterized complexity: "In the beginning, all we did was graph minors" (M. Fellows, pers. comm.). As such, the impact of the theory of sparse graphs on algorithmic research cannot be overstated.

Many of the algorithmic results concerning classes excluding a minor or a topological minor are in some way based on topological arguments, depending on the structure theorems (e.g. decompositions) for the class under consideration. A complete paradigm shift was initiated by Nešetřil and Ossona de Mendez with their foundational work and introduction of the notions of *bounded expansion* [42, 43, 44] and *nowhere denseness* [46]. These graph classes extend and properly contain all the aforementioned sparse classes and many arguments based on topology can be replaced by more general, and surprisingly often much simpler, arguments based on *density*. We refer to the textbook [47] for extensive background on the theory of sparse graph classes.

The rich structural theory for bounded expansion and nowhere dense graph classes has been successfully applied to design efficient algorithms for hard computational problems on specific sparse classes of graphs, see e.g. [6, 16, 21, 22, 23, 24, 25, 28, 30, 63]. On the other hand, several results indicate that nowhere dense graph classes form a natural limit for algorithmic methods based on sparseness arguments, see e.g. [21, 23].

One core strength of the bounded expansion/nowhere dense framework is that there exists a multitude of equivalent definitions that provide complementing perspectives. Here, we study two structural properties of these classes that are of particular importance in the algorithmic context, namely the property of having bounded *generalized coloring numbers* and the property of being *uniformly quasi-wide*. The generalized coloring numbers intuitively measure reachability properties in a linear vertex ordering of a given graph. Such an ordering yields a very weak and local form of a graph decomposition which can be exploited combinatorially [24, 54] and algorithmically [6, 21, 22, 30]. Uniform quasi-wideness was originally introduced in finite model theory [15], and soon found combinatorial and algorithmic applications on nowhere dense classes [16, 24, 28, 35, 45, 52, 60].

Even though the above results render many problems tractable in theory, many of the known algorithms have worst-case running times that involve huge constant factors and combinatorial explosions with respect to the discussed parameters. The central question of our work here is to investigate how the generalized coloring numbers and uniform quasi-wideness behave on real-world graphs, an endeavor which so far has only been conducted

for a single notion of bounded expansion and on a smaller scale [20]. Controllable numbers would be a prerequisite for practical implementations of these algorithms based on such structural approaches. We provide an experimental evaluation of several algorithms that approximate these parameters on real world graphs.

On the theoretical side, we provide a new algorithm for uniform quasi-wideness with polynomial size guarantees in graph classes of bounded expansion and show a lower bound indicating that the guarantees of this algorithm are close to optimal in graph classes with fixed excluded minor.

2 Basic definitions

Graphs. All graphs in this paper are finite, undirected and simple, that is, they do not have loops or multiple edges between the same pair of vertices. For a graph G, we denote by $V(G)$ the vertex set of G and by $E(G)$ its edge set. The *distance* between a vertex v and a vertex w is the length (that is, the number of edges) of a shortest path between v and w. For a vertex v of G, we write $N^G(v)$ for the set of all neighbors of v, $N^G(v) = \{ u \in V(G) \mid \{u, v\} \in E(G) \}$, and for $r \in \mathbb{N}$ we denote by $N^G_r[v]$ the *closed r-neighborhood of v*, that is, the set of vertices of G at distance at most r from v. Note that we always have $v \in N^G_r[v]$. The radius of a connected graph G is the minimum integer r such that there exists $v \in V(G)$ with the property that all vertices of G have distance at most r to v. A set A is *r-independent* if all distinct vertices of A have distance greater than r.

Bounded expansion and nowhere denseness. A *minor model* of a graph H in a graph G is a family $(I_u)_{u \in V(H)}$ of pairwise vertex-disjoint connected subgraphs of G, called *branch sets*, such that whenever uv is an edge in H, there are $u' \in V(I_u)$ and $v' \in V(I_v)$ for which $u'v'$ is an edge in G. The graph H is a *depth-r minor* of G, denoted $H \preccurlyeq_r G$, if there is a minor model $(I_u)_{u \in V(H)}$ of H in G such that each I_u has radius at most r. A class $\mathcal{C}$ of graphs is *nowhere dense* if there is a function $t\colon \mathbb{N} \to \mathbb{N}$ such that for all $r \in \mathbb{N}$ it holds that $K_{t(r)} \not\preccurlyeq_r G$ for all $G \in \mathcal{C}$, where $K_{t(r)}$ denotes the clique on $t(r)$ vertices. The class $\mathcal{C}$ has *bounded expansion* if there is a function $d\colon \mathbb{N} \to \mathbb{N}$ such that for all $r \in \mathbb{N}$ and all $H \preccurlyeq_r G$ with $G \in \mathcal{C}$, the *edge density* of H, i.e. $|E(H)|/|V(H)|$, is bounded by $d(r)$.

Weak coloring numbers. The *weak coloring numbers* wcol_r were introduced by Kierstead and Yang [31] and intuitively measure reachability properties in a linear vertex ordering of a given graph. Formally, they are a series of numbers, parameterized by a positive integer r, which denotes the radius of the considered ordering. Let $\Pi(G)$ be the set of all linear orders of the vertices of the graph G, and let $L \in \Pi(G)$. Let $u, v \in V(G)$. For a positive integer r, we say that u is *weakly r-reachable* from v with respect to L, if there exists a path P of length ℓ, $0 \leq \ell \leq r$, between u and v such that u is minimum among the vertices of P (with respect to L). Let $\mathrm{WReach}_r[G, L, v]$ be the set of vertices that are weakly r-reachable from v with respect to L. Note that $v \in \mathrm{WReach}_r[G, L, v]$. The *weak r-coloring number* $\mathrm{wcol}_r(G)$ of G is defined as

$$\mathrm{wcol}_r(G) := \min_{L \in \Pi(G)} \max_{v \in V(G)} \left|\mathrm{WReach}_r[G, L, v]\right|.$$

As proved by Zhu [67], the weak coloring numbers can be used to characterize bounded expansion and nowhere dense classes of graphs.

Uniform quasi-wideness. Intuitively, a class of graphs is *wide* if for every graph G from the class, every radius $r \in \mathbb{N}$ and every large subset $A \subseteq V(G)$ of vertices one can find a large r-independent subset $B \subseteq A$. The notion of uniform quasi-wideness allows to additionally delete a small number of vertices to make B r-independent. The following definition formalizes the meaning of "large" and "small".

▶ **Definition 2.1.** A class $\mathcal{C}$ of graphs is *uniformly quasi-wide* if for every $m \in \mathbb{N}$ and every $r \in \mathbb{N}$ there exist numbers $N(m,r)$ and $s(r)$ such that the following holds.

> *Let $G \in \mathcal{C}$ and let $A \subseteq V(G)$ with $|A| \geq N(m,r)$. Then there exists a set $S \subseteq V(G)$ with $|S| \leq s(r)$ and a set $B \subseteq A \setminus S$ of size at least m such that for all distinct $u, v \in B$ we have $\mathrm{dist}_{G-S}(u,v) > r$.*

Uniform quasi-wideness was introduced by Dawar in [15] and it was proved by Nešetřil and Ossona de Mendez in [45] that uniform quasi-wideness is equivalent to nowhere denseness.

3 Weak coloring numbers

We experiment with the following approximation algorithms of weak coloring numbers. We here only briefly list them and give necessary definitions to discuss studied variants; a more exhaustive presentation can be found in the full version of the paper.

Distance-constrained transitive fraternal augmentations. We can approximate the weak coloring numbers by orienting the input graph G and iteratively inserting arcs according to certain rules. Such *transitive-fraternal augmentations* (tf-augmentations) were studied first in [43]. We work with an optimized version, called *distance-constrained tf-augmentations* (dtf-augmentations) which were introduced in [53].

Flat decompositions. The following algorithm was introduced in [62]. It provides a way of constructing an order with bounded $wcol_r$ numbers on class of graphs with excluded minors.

Consider the following procedure for computing a vertex ordering of G. At each step, we maintain a family of *blobs* $B_1, B_2, \ldots, B_p \subseteq V(G)$, which are pairwise disjoint and connected, and we let $U := V(G) \setminus \bigcup_{i=1}^{p} B_i$ be the vertices which are not yet contained in any blob. We call vertices in U *unprocessed* and vertices in $V(G) \setminus U$ *processed*. To create the next blob, we let u be any vertex of U and let C be the connected component of $G[U]$ that contains u. Create blob B_{p+1} as follows: start with $\{u\}$, and for every blob B_i that is adjacent to C, pick any vertex $v \in C$ adjacent to B_i, and add to B_{p+1} any shortest path from u to v within C. Finally, when all vertices are subsumed in the blobs, order vertices from different blobs according to the creation time of their blobs, and vertices from the same blob arbitrarily.

As shown in [62], if $K_t \not\preccurlyeq G$, then the above procedure produces an order that certifies that $\mathrm{wcol}_r(G) \in O(r^{t-1})$. Note that this algorithm leaves a lot of room for heuristic optimizations: we can first vary the order of vertices within the blobs and we can vary the choice of the vertex u. As it is not clear which choices would be the best, we decided to create a few sets of rules for both choices and evaluate every combination of them. Within one blob we can order vertices (1) according to a BFS, (2) according to a DFS, (3) in the order of descending degree (motivated by the results of another heuristic). In the tables presented in Section 6, these rules will be abbreviated as BFS, DFS and SORT, respectively. Moreover, each of these orders can be reversed; reversed orders are denoted with an overline over their acronym.

As the next unprocessed vertex u we can choose a vertex (1) with the largest number of processed neighbours, (2) with the largest degree among all unprocessed vertices, (3) with the largest degree among all unprocessed vertices with a processed neighbor. Later, we refer to these rules by their numbers.

Treedepth heuristic. Since the 'limit' of weak-coloring numbers is exactly the treedepth of a graph, i.e. $\mathrm{wcol}_\infty(G) = \mathrm{td}(G)$, we consider computing a treedepth decomposition and using an ordering derived from the decomposition. Our algorithm of choice, developed by Sanchez [59] and implemented by Oelschlägel [51], recursively extracts separators from the graph.

Treewidth heuristic. A well-known approach to compute a treewidth decomposition of a graph is to find a linear order of the vertices, an elimination order, of possibly small maximum so-called "back-degree". There is a number of heuristics to produce good elimination orders. We chose one that is simple, fast and that gives rather good results for treewidth: the so-called minimum-degree heuristic [9].

Other simple heuristics. Apart from algorithms with theoretical guarantees we also compared several naive heuristics.

- For $r = 1$ an optimal order is a degeneracy order, which can be easily computed. We can check if this order produces reasonable results for higher values of r as well.
- Intuitively, it makes sense to sort vertices by descending degree (ties are broken arbitrarily) because from vertices of high degree more vertices can be reached in one step.
- A simple idea of generalizing the above heuristics to bigger values of r is to apply them to the rth power G^r of G (G^r is defined as the graph with $V(G^r) = V(G)$ and $uv \in E(G^r) \Leftrightarrow \mathrm{dist}_G(u, v) \leq r$).
- As a baseline we also included random ordering of vertices.

The intuition behind using a degree-ordering is further supported by a popular network model: *Chung–Lu* random graphs which sample graphs with a fixed degree distribution and succesfully replicate several statistics exhibited by real-world networks [12, 13]. In this model, vertices are assigned weights (corresponding to their expected degree) and edges are sampled independently but biased according to the endpoints weights. Therefore vertices of the same degree are exchangable and the one ordering we can choose to minimize the number of r-reachable vertices is simply the descending degree ordering. It follows that if Chug–Lu graphs are a resonable approximation of real-world networks, then the degree ordering should a good choice.

3.1 Local search

In addition to all these approaches we can try to improve their results by local search, a technique where we make small changes to a candidate solution. We applied the following local changes and tested whether they caused improvements to the current order L.

- Swap a vertex v that has biggest $\mathrm{WReach}_r[G, L, v]$ with a random vertex that is smaller with respect to L.
- Swap a vertex v that has biggest $\mathrm{WReach}_r[G, L, v]$ with its direct predecessor u in L.

Both heuristics try to place a vertex with many weakly reachable vertices earlier in the order and thus to make them non-weakly reachable. The advantage of the second rule is that $\mathrm{WReach}_r[G, L, v]$ is trivial to recompute and the only computationally heavy update is for the new $\mathrm{WReach}_r[G, L, u]$. For the first rule, recomputing WReach sets is more expensive. However, the disadvantage of the second rule is that it does not lead to further improvements quickly, hence applications of only the first rule give better results than applications of the second rule only. In our implementation we did a few optimizations in order to improve the results of second rule, but we refrain from describing them in detail. The final algorithm conducting local search firstly performs a round of applications of the first rule and when they no longer improve results it performs a round of applications of the second rule. Such combination turned out to be empirically most effective.

4 Uniform quasi-wideness

We experiment with the following algorithms for uniform quasi-wideness. We here only briefly list them and give necessary definitions to discuss studied variants; more exhaustive presentation can be found in the full version of the paper.

Distance trees. [52] introduced a method for showing uniform quasi-wideness of nowhere dense graphs by iteratively building r-independent sets for increasing values of r. The critical part is an algorithm that, given an (1-)independent set A in a graph G, finds a (small) set S and a 2-independent set $B \subseteq A$ in $G - S$. An involved combinatorial argument shows the following: either such set B can be already found for the tentative S, or there exists a vertex $v \in V(G)$ with many neighbors in A; then one includes v in S and restrict A to $N(v) \cap A$. The final restriction is critical for the proof of the bound on the final set S.

We have implemented three variants of this algorithm, denoted later `tree1`, `tree2`, and `ld_it`. `tree2` is the original algorithm of [52], while `tree1` is a variant that, in the step when the set A is restricted to $N(v) \cap A$, tries to preserve some vertices of $A \setminus N(v)$ for future use. Finally, `ld_it` is a variant that replaces every execution of the method of [52] with greedy approach to search for large 2-independent set $B \subseteq A$.

From weak coloring numbers to uniform quasi-wideness. First, we implemented an approach of [34] which is designed for classes of bounded expansion and combines the weak coloring numbers with uniform quasi-wideness. This algorithms is later referred to as `mfcs`.

Second, motivated by the rather conservative character of the algorithm of [34], we propose here a new algorithm (albeit inspired by [34]), proving the following.

▶ **Theorem 4.1.** *Assume we are given a graph G, a set $A \subseteq V(G)$, integers $r \geq 1$ and $m \geq 2$, and an ordering L of $V(G)$ with $c = \max_{v \in V(G)} |\mathrm{WReach}_r[G, L, v]|$. Furthermore, assume that $|A| \geq 4 \cdot (2cm)^c$. Then in polynomial time, one can compute sets $S \subseteq V(G)$ and $B \subseteq A \setminus S$ such that $|S| \leq c$, $|B| \geq m$, and B is r-independent in $G - S$.*

We implemented three variants of the above algorithm, `new1`, `new2`, and `new_ld`. The first two differ in some minor internal details, whereas `new_ld` extends `new2` as follows: at every step it attempts to complete the currently handled partial r-independent set in a greedy manner, and at the end returns the best solution found during the entire execution.

Other naive approaches and heuristic optimizations. Since computing uniform quasi-wideness for $r = 1$ is equivalent to finding independent sets, it is sensible to include independent set heuristics as a baseline. Moreover, the approach based on distance trees computes independent sets as a subroutine. We used a simple greedy algorithm to find independent sets: As long as our graph is nonempty, take a vertex of minumun degree, add it to the independent set and remove its closed neighborhood from the graph.

The following algorithm is what we came up with as a naive but reasonable heuristic for larger values of r. For every number $k \in \{0, 1, \ldots, K\}$ (where K is some hardcoded constant) compute the biggest independent set in graph $(G - S_k)^r[A]$ using the greedy procedure described above, where S_k is a set of k vertices with biggest degrees. This heuristic is based on the fact that independent sets in G^r correspond to r-independent sets in G. Without any further knowledge about the graph, vertices with the biggest degree seem to be the best candidates to be removed. In the end, we output the best solution obtained in this manner. In the following, we abbreviate this approach as `ld` (least degree on power graph).

4.1 Score: Comparing different results

Uniform quasi-wideness is a two-dimensional measure: we have to measure both the size m of the r-independent set B which we desire to find, as well as the size $s(r)$ of vertices to be deleted. In order to compare the performance of our studied methods we propose the following approach that arises from applications of uniform quasi-wideness in several algorithms [16, 21, 24, 52, 60].

Let $G, A \subseteq V(G), r \in \mathbb{N}$ be an input to any of our algorithms (note that none of our algorithms takes the target size m of the r-independent set as input, we rather try to maximize its size) and let $S \subseteq V(G)$ and $B \subseteq A \setminus S$ such that B is r-independent in $G - S$ be its output. Let us define $\pi_r[v, S]$ – the *r-distance profile of v on S* – as the function from S to $\{0, 1, \ldots, r, \infty\}$ so that $\pi_r[v, S](a) = dist_G(v, a)$ if this distance is at most r, and $\pi_r[v, S](a) = \infty$ otherwise. The performance of the algorithms [16, 21, 24, 52, 60] strongly depends on the size of the largest equivalence class on B defined by $u \sim v$ if $\pi_r[u, S] = \pi_r[v, S]$ for $u, v \in B$.

We hence decided to use the size of the largest equivalence class in the above relation as the scoring function to measure the performance of our algorithms. Note that number of different r-distance profiles is bounded by $(r + 2)^{|S|}$, so if r is fixed and $|S|$ is bounded then the number of different r-distance profiles is also bounded, so having a big r-independent set implies having a big subset of this set with equal r-distance profiles on S.

This well defined scoring function makes it possible to compare the results of the algorithms. Furthermore, in our code the implementation of the scoring function can be easily exchanged, so if different scoring functions are preferred, re-evaluation is easily possible.

5 Experimental setup

5.1 Hard- and Software

The experiments on generalized coloring numbers has been performed on an Asus K53SC laptop with Intel® Core™ i3-2330M CPU @ 2.20GHz x 2 processor and with 7.7GiB of RAM. Weak coloring numbers of a larger number of graphs for the statistics in Section 6.4 (presented without running times) were produced on a cluster at the Logic and Semantics Research Group, Technische Universität Berlin. The experiments on uniform quasi-wideness have been performed on a cluster of 16 computers at the Institute of Informatics, University of Warsaw. Each machine was equipped with Intel Xeon E3-1240v6 3.70GHz processor and 16 GB RAM. All machines shared the same NFS drive. Since the size of the inputs and outputs to the programs is relatively small, the network communication was neglible for tests with substantial running times. The dtf implementation has been done in Python, while all other code in C++ or C. The code is available at [41, 3].

5.2 Test data

Our dataset consists of a number of graphs from different sources.

Real-world data. We collected appropriately-sized networks from several collections [1, 33, 39, 7, 58, 36]. Our selection contains classic social networks [66, 11], collaboration networks [38, 49, 48] contact networks [61, 40], communication patterns [38, 56, 32, 37, 55, 4], protein-protein interaction [10], gene expression [27], infrastructure [64], tournament data [26], and neural networks [65]. We kept the names assigned to these files by the respective source.

PACE 2016 Feedback Vertex Set. The Parameterized Algorithms and Computational Experiments Challenge is an annual programming challenge started in 2016 that aims at *investigate the applicability of algorithmic ideas studied and developed in the subfields of multivariate, fine-grained, parameterized, or fixed-parameter tractable algorithms* (from the PACE webpage). In the first edition, one of the tracks focused on the Feedback Vertex Set problem [17], providing 230 instances from various sources and of different sizes. We have chosen a number of instances with small feedback vertex set number, guaranteeing their very strong sparsity properties (in particular, low treewidth). In our result tables, they are named `fvs???`, where `???` is the number in the PACE 2016 dataset.

Random planar graphs. In their seminal paper, Alber, Fellows, and Niedermeier [5] initiated the very fruitful direction of developing of polynomial kernels (preprocessing routines rigorously analyzed through the framework of parameterized complexity) in sparse graph classes by providing a linear kernel for Dominating Set in planar graphs. In [5], an experimental evaluation is conducted on random planar graphs generated by the LEDA library [2]. We followed their setup and included a number of random planar graphs with various size and average degree. In our result tables, they are named `planarN`, where `N` stands for the number of vertices.

Random graphs with bounded expansion. A number of random graph models has been shown to produce almost surely graphs of bounded expansion [20]. We include a number of graphs generated by O'Brien and Sullivan [50] using the following models: the stochastic block model (`sb-?` in our dataset) [29] and the Chung-Lu model with households (`clh-?`) and without households (`cl-?`) [14]. We refer to [20, 50] for more discussion on these sources.

The graphs have been partitioned into four groups, depending on their size: the `small` group gathers graphs up to 1 000 edges, `medium` between 1 000 and 10 000 edges, `big` between 10 000 and 48 000 edges, and `huge` above 48 000 edges. The random planar graphs in every test group have respectively 900, 3 900, 21 000, and 150 000 edges. The whole dataset is available for download at [3].

6 Weak coloring numbers: results

6.1 Fine-tuning flat decompositions

As discussed in Section 3, we have experimented with a number of variants of the flat decompositions approach, with regards to the choice of the next root vertex and the internal order of the vertices of the next B_i. The results for the `big` dataset are presented in Table 1. They clearly indicate that (a) all reversed orders performed much worse, and (b) among other options, the best is to sort the vertices of a new B_i nonincreasingly by degree and choose as the next root the vertex of maximum degree. In the subsequent tests, we use this best configuration for comparison with other approaches.

6.2 Comparison of all approaches

Table 2 presents the results of our experiments on all test instances and all approaches, summarized as follows:

dtf dtf-augmentations with the respective radius r supplied as the distance bound;
flat the best configuration of the flat decompositions approach (see previous section);
treedepth the treedepth approximation heuristic;
treewidth the treewidth heuristic;

Table 1 Comparison of different flat decomposition variants: sorting vertices of the new blobs B_i by the BFS, DFS, by degree (nonincreasing), or these orders reversed; the second coordinate refers to the choice of the root vertex: (1) maximizing the number of neighbors already processed, (2) maximizing degree in U, (3) as previous, but only among neighbors of already processed vertices. The value is the average of the approximation ratios to the best generalized coloring numbers found by all versions of this algorithm.

option	average appx. ratio	option	average appx. ratio	option	average appx. ratio
BFS/(1)	1.159	DFS/(1)	1.156	SORT/(1)	1.072
BFS/(2)	1.131	DFS/(2)	1.117	**SORT/(2)**	**1.039**
BFS/(3)	1.147	DFS/(3)	1.135	SORT/(3)	1.054
$\overline{\text{BFS}}$/(1)	1.363	$\overline{\text{DFS}}$/(1)	1.368	$\overline{\text{SORT}}$/(1)	1.41
$\overline{\text{BFS}}$/(2)	1.277	$\overline{\text{DFS}}$/(2)	1.291	$\overline{\text{SORT}}$/(2)	1.329
$\overline{\text{BFS}}$/(3)	1.309	$\overline{\text{DFS}}$/(3)	1.324	$\overline{\text{SORT}}$/(3)	1.36

Table 2 *Gray columns*: Comparison of the main approaches and their average approximation ratio to the best found coloring number. Some of the approaches did not finish in time on larger graphs or ran out of memory. *White columns*: Total running time of the main approaches. Note that for some approaches the ordering (and thus running time) is independent of the radius.

tests	r	dtf		flat		treedepth		treewidth		degree sort	
small	2	1.19	0:04.20	1.2	0:00.16	1.408	0:08.97	1.12	0:00.34	1.179	0:00.09
	3	1.439	0:05.08	1.239		1.438		1.124		1.211	
	4	1.558	0:05.74	1.288		1.384		1.135		1.213	
	5	1.718	0:06.55	1.353		1.414		1.167		1.263	
medium	2	1.177	0:27.97	1.362	0:01.97	2.171	—	1.524	0:23.64	1.142	0:00.56
	3	1.258	1:02.31	1.43		1.918		1.283		1.102	
	4	1.499	1:53.21	1.451		1.698		1.159		1.113	
	5	1.595	2:15.04	1.469		1.612		1.093		1.149	
big	2	1.107	0:32.82	1.43	0:19.08	—	—	2.278	—	1.183	0:03.30
	3	1.243	—	1.419		—		1.895		1.088	
	4	—	—	1.414		—		1.434		1.079	
	5	—	—	1.415		—		1.189		1.065	
huge	2	—	—	1.727	—	—	—	—	—	1.152	—
	3	—	—	2.156		—		—		1.031	
	4	—	—	2.13		—		—		1.032	
	5	—	—	2.095		—		—		1.029	

degree sort the heuristic which sorts the vertices nonincreasing by degree.

Out of all simple heuristics (c.f. Section 3) the degree sorting was supreme and we skip the results of inferior heuristics (see [41, 3] for full data). Interestingly, this heuristic also outperformed all other (much more involved) approaches on larger graphs. On small graphs, the treewidth heuristic takes the lead. An explanation why the treewidth heuristic is better on smaller graphs G might be that $\mathrm{tw}(G) = \mathrm{col}_\infty(G)$ and on small graphs the difference between $\mathrm{col}_\infty(G)$ and $\mathrm{col}_r(G)$ for the considered r is not that big. However, this does not explain why treedepth does not perform better than treewidth. (Recall that $\mathrm{td}(G) = \mathrm{wcol}_\infty(G)$.) It is worth observing that on larger graphs (the `big` group) the performance of the flat

Table 3 *Gray columns*: Comparison of average approximation ratio after local search. *White columns*: Relative improvement of local search for ordering output by the studied approaches.

tests	radius	dtf		flat		treedepth		treewidth		degree sort	
small	2	1.126		1.032		1.142		1.059		1.025	
	3	1.227	16.7%	1.076	16.9%	1.235	15.2%	1.098	7.0%	1.044	16.2%
	4	1.327		1.091		1.281		1.131		1.053	
	5	1.466		1.135		1.311		1.154		1.088	
medium	2	1.192		1.138		1.206		1.135		1.011	
	3	1.204	13.9%	1.115	21.4%	1.303	30.9%	1.121	15.3%	1.023	17.1%
	4	1.444		1.28		1.349		1.139		1.017	
	5	1.482		1.325		1.401		1.134		1.034	
big	2	1.12		1.142		—		1.201		1.045	
	3	1.218	—	1.14	24.4%	—	—	1.29	24.3%	1.015	18.3%
	4	—		1.223		—		1.27		1.017	
	5	—		1.257		—		1.212		1.022	

decomposition matches or outperforms the one of the treewidth heuristic for radii $r = 2, 3, 4$. However, the treewidth heuristic outperforms all approaches with proved guarantees for $r = 5$ on test sets up to the `big` group.

Table 2 gathers total running time of our programs on discussed data sets. These results clearly indicate large discrepancy between consumed resources for different approaches. Out of the approaches with provable guarantees on the output coloring number, the flat decompositions approach is clearly the most efficient.

Note that we applied different timeout policies for generating different data. For generating time of execution and for applying local search we set timeout to be 1 minute, however for generating orders and wcol numbers we set timeout to be 5 minutes, but for the sake of completeness we sometimes allowed some programs to run longer.

In summary, on our data sets the simple heuristic is consistently the fastest and produces the best results, save for the smallest graphs on which the treewidth heuristic won. We remark here that it is simple to "fool" the degree-sorting heuristic by adding multiple pendant vertices of degree one and thus forcing it to take arbitrarily bad ordering, but such adversarial obstacles seem to be absent in real-world graphs. If one is to choose an algorithm with provable guarantees, the discussed variant of the flat decompositions approach appears to be the best choice.

6.3 Local search

In a second round of experiments we applied a simple local-search routine that, given an ordering output by one of the approaches, tries to improve it by moving vertices with the largest weakly reachable sets earlier in the ordering. The white columns in Table 3 show how local search improved orderings output by discussed approaches, and the gray columns show average approximation ratios of orderings improved by local search. Two remarks are in place. First, regardless of how the ordering was computed, a local search step always significantly improves the ordering (we have no good explanation on why local search is significantly less effective on the orderings output by the treewidth heuristic for bigger radii). Second, the local search step does *not* improve the orderings enough to change the relative order of the performance of the base approaches except for one remarkable case. On `medium` group the treewidth heuristic gave best results on $r = 5$, however degree sort regained the lead after

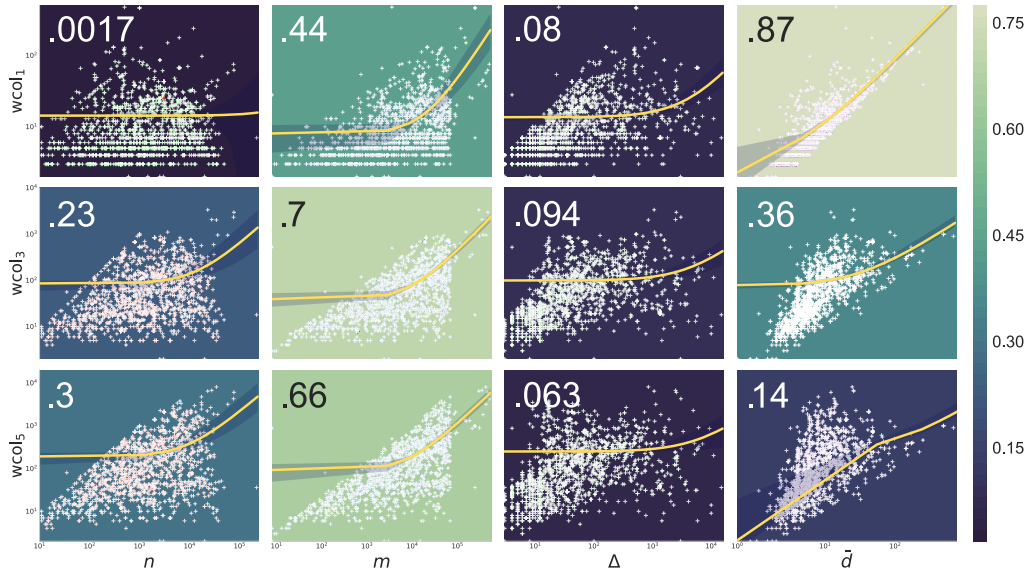

Figure 1 Correlation of wcol (computed using the degree sort heuristic) with graph size, maximum degree and average degree of 1675 real-world graphs. The background shade and number reflect the correlation of the two respective measures, superimposed is a log-log plot of the measurements. The yellow lines are linear regressions with dark shaded confidence intervals.

application of local search due to its low performance on larger radii for treewidth heuristic. We therefore recommend the local search improvement as a relatively cheap post-processing improvement to any existing algorithm.

6.4 Correlation of weak coloring numbers with other parameters

While it is undeniable that weak coloring numbers have immense algorithmic power from a theoretical perspective, the efficient computation of such weak coloring orders is only one component to leverage them in practice: we also need these numbers to be reasonably low. So far, this had only been established on a smaller scale [20, 53] for a related measure. Here, we computed the weak coloring number for $r \in \{1, \ldots, 5\}$ for 1675 real-world networks from various sources [36, 39, 58, 7, 1]. Figure 1 summarizes our findings: for $r \in \{1, \ldots, 3\}$ we find a modest correlation with n and a significant correlation with m. The correlation with n becomes quite pronounced for $r = 5$; the probable reason being that for all networks involved $\log n \leq 10$. Still, even in the worst examples wcol_5 is at least one order of magnitude smaller than n or m. We further see a high correlation between wcol_1 and the average degree $\bar{d}$ which vanishes for larger radii. It is no big surprise that $\bar{d}$ and the degeneracy wcol_1 are highly correlated since these values are only far apart in graphs with highly inhomogeneous densities.

The low dependence on the maximum degree confirms the findings of [20]: the exact shape of the degree distribution's tail is much more relevant than the singular value of the maximum degree. Finally, note that in our graphs the degeneracy wcol_1 practically does not grow with n.

Table 4 Aggregated results of uniform quasi-wideness on `medium` set for $r = 3$ and $r = 5$ (values for $r = 2$ and $r = 4$ can be found in the full version of the paper): total size of all deleted and independent sets, total score (total size of largest equivalance classes w.r.t. deleted vertices), and total running time.

r	algorithm	start with whole $V(G)$				start with 20% of $V(G)$			
		deleted	independent	score	time	deleted	independent	score	time
3	`mfcs`	5076	11471	2153	0:01.25	1922	3459	1135	0:00.48
	`new1`	78	2345	2211	0:37.53	49	1192	1159	0:29.96
	`new2`	84	3820	3673	0:34.34	49	2132	2096	0:23.36
	`new_ld`	—	—	—	—	5	2926	2873	11:10.63
	`tree1`	7	6072	5686	0:02.77	4	2652	2598	0:00.48
	`tree2`	5	5645	5645	0:01.00	4	2603	2603	0:00.38
	`ld_it`	7	6136	5748	0:01.71	4	2741	2688	0:00.39
	`ld`	5	6471	6296	0:08.13	6	2972	2871	0:02.01
5	`mfcs`	7946	15773	1164	0:01.93	4057	4396	594	0:00.67
	`new1`	115	1623	1445	4:38.57	84	709	676	3:20.15
	`new2`	122	2079	1888	4:19.50	103	1036	982	3:07.82
	`new_ld`	—	—	—	—	—	—	—	—
	`tree1`	11	2988	2643	0:02.85	4	1325	1282	0:00.53
	`tree2`	5	2603	2603	0:01.05	4	1284	1284	0:00.45
	`ld_it`	12	3102	2752	0:01.84	5	1380	1336	0:00.64
	`ld`	7	3192	3043	0:29.32	5	1517	1473	0:07.15

7 Uniform quasi wideness: results

Table 4 gathers aggregated data from our experiments on `medium` dataset. (Full data can be downloaded from [41, 3].) Every tested algorithm has been run on every test with timeout 10 minutes and with radii $r \in \{2, 3, 4, 5\}$ and with the starting set either $A = V(G)$ or a random subset of 20% of vertices of $V(G)$.

Data indicate the simple heuristic, `ld`, as the best choice in most scenarios, as it has always best or nearly-best total score and runs relatively quickly. The third variant of the new algorithm `new_ld` has comparable results, but is inefficient and does not finish within the timeout. Other variants `new1` and `new2` as well as `mfcs` are significantly outperformed by other approaches. Out of other approaches with provable guarantees, the variants `tree1`, `tree2`, and `ld_it` provide results in most cases less than 10% worse than the heuristic `ld`, with `tree2` being consistently worse.

To sum up, our experiments show that the simple heuristic `ld` gives best results, but if one is interested in algorithm with provable guarantees, one should choose one of the variant `tree1` over `mfcs` or `new1`/`new2`.

8 Conclusions

We have conducted a thorough empirical evaluation of algorithms for computing generalized coloring numbers and uniform quasi-wideness. In both cases, one of the simplest heuristics, without any theoretical guarantees, outperformed the rest. In particular, our new algorithm for uniform quasi-wideness, whose development was motivated by conservativeness of the previous approach of [34], performed rather poorly in the experiments. From the algorithms

with provable guarantees, the experiments indicated a variant of the algorithm of [62] as the algorithm of choice for generalized coloring numbers and a variant of the algorithm of [52] as the algorithm of choice for uniform quasi-wideness.

References

1 Gephi datasets. URL: https://github.com/gephi/gephi/wiki/Datasets.

2 LEDA. URL: http://www.algorithmic-solutions.com/leda/index.htm.

3 Recent trends in kernelization: theory and experimental evaluation — project website, 2018. URL: http://kernelization-experiments.mimuw.edu.pl.

4 Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 US election: divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, pages 36–43. ACM, 2005.

5 Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *Journal of the ACM*, 51(3):363–384, 2004. doi:10.1145/990308.990309.

6 Saeed Akhoondian Amiri, Patrice Ossona de Mendez, Roman Rabinovich, and Sebastian Siebertz. Distributed domination on graph classes of bounded expansion. *CoRR*, abs/1702.02848, 2017.

7 Vladimir Batagelj and Andrej Mrvar. Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/, 2006.

8 Hans L. Bodlaender. Treewidth: Algorithmic techniques and results. In *International Symposium on Mathematical Foundations of Computer Science*, pages 19–36. Springer, 1997.

9 Hans L. Bodlaender and Arie M. C. A. Koster. Treewidth computations I. Upper bounds. *Information and Computation*, 208(3):259–275, 2010. doi:10.1016/j.ic.2009.03.008.

10 Dongbo Bu, Yi Zhao, Lun Cai, Hong Xue, Xiaopeng Zhu, Hongchao Lu, Jingfen Zhang, Shiwei Sun, Lunjiang Ling, Nan Zhang, et al. Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic acids research*, 31(9):2443–2450, 2003.

11 Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1082–1090, 2011. doi:10.1145/2020408.2020579.

12 Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

13 Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2):125–145, 2002.

14 Fan R. K. Chung and Linyuan Lu. The average distance in a random graph with given expected degrees. *Internet Mathematics*, 1(1):91–113, 2003. doi:10.1080/15427951.2004.10129081.

15 Anuj Dawar. Homomorphism preservation on quasi-wide classes. *Journal of Computer and System Sciences*, 76(5):324–332, 2010.

16 Anuj Dawar and Stephan Kreutzer. Domination problems in nowhere-dense classes. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, pages 157–168, 2009.

17 Holger Dell, Thore Husfeldt, Bart M. P. Jansen, Petteri Kaski, Christian Komusiewicz, and Frances A. Rosamond. The First Parameterized Algorithms and Computational Experiments Challenge. In *11th International Symposium on Parameterized and Exact Computation (IPEC 2016)*, volume 63 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:9, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.IPEC.2016.30.

18 Erik D. Demaine and MohammadTaghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3):292–302, 2007.

19 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Improved grid minor bounds and wagner's contraction. *Algorithmica*, 54(2):142–180, 2009.

20 Erik D. Demaine, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, Somnath Sikdar, and Blair D. Sullivan. Structural sparsity of complex networks: Random graph models and linear algorithms. *CoRR*, abs/1406.2587, 2014. `arXiv:1406.2587`.

21 Pål Grønås Drange, Markus Sortland Dregi, Fedor V. Fomin, Stephan Kreutzer, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, Felix Reidl, Fernando Sánchez Villaamil, Saket Saurabh, Sebastian Siebertz, and Somnath Sikdar. Kernelization and sparseness: the case of dominating set. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS*, pages 31:1–31:14, 2016.

22 Zdenek Dvorak. Constant-factor approximation of the domination number in sparse graphs. *European Journal of Combinatorics*, 34(5):833–840, 2013.

23 Zdenek Dvorak, Daniel Král, and Robin Thomas. Testing first-order properties for subclasses of sparse graphs. *Journal of the ACM*, 60(5):36:1–36:24, 2013.

24 Kord Eickmeyer, Archontia C. Giannopoulou, Stephan Kreutzer, O-joung Kwon, Michał Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. Neighborhood complexity and kernelization for nowhere dense classes of graphs. In *44th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 63:1–63:14, 2017.

25 Jakub Gajarský, Petr Hliněný, Jan Obdržálek, Sebastian Ordyniak, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. Kernelization using structural parameters on sparse graph classes. *Journal of Computer and System Sciences*, 84:219–242, 2017.

26 Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

27 Kwang-Il Goh, Michael E. Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

28 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM*, 64(3):17:1–17:32, 2017.

29 Paul W. Holland, Kathryn B. Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

30 Wojciech Kazana and Luc Segoufin. Enumeration of first-order queries on classes of structures with bounded expansion. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS*, pages 297–308, 2013.

31 Henry A. Kierstead and Daqing Yang. Orderings on graphs and game coloring number. *Order*, 20:255–264, 2003.

32 Bryan Klimt and Yiming Yang. Introducing the Enron corpus. In *CEAS*, 2004.

33 Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*, volume 37. Addison-Wesley Reading, 1993.

34 Stephan Kreutzer, Michał Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. The generalised colouring numbers on classes of bounded expansion. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS*, 2016.

35 Stephan Kreutzer, Roman Rabinovich, and Sebastian Siebertz. Polynomial kernels and wideness properties of nowhere dense graph classes. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1533–1545, 2017.

36 Jérôme Kunegis. KONECT - the Koblenz network collection. In *Proc. Int. Web Observatory Workshop*, pages 1343–1350, 2013. URL: `http://konect.uni-koblenz.de`.

37 Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370. ACM, 2010.

38 Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

39 Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, 2014.

40 David Lusseau, Karsten Schneider, Oliver J. Boisseau, Patti Haase, Elisabeth Slooten, and Steve M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

41 Wojciech Nadara, Marcin Pilipczuk, Felix Reidl, Roman Rabinovich, and Sebastian Siebertz. Empirical evaluation of approximation algorithms for generalized graph coloring and uniform quasi-wideness. code repository, 2018. `https://bitbucket.org/marcin_pilipczuk/wcol-uqw-experiments`.

42 Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion I. decompositions. *European Journal of Combinatorics*, 29(3):760–776, 2008.

43 Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion II. algorithmic aspects. *Europeean Journal of Combinatorics*, 29(3):777–791, 2008.

44 Jaroslav Nešetřil and Patrice Ossona de Mendez. Grad and classes with bounded expansion III. restricted graph homomorphism dualities. *European Journal of Combinatorics*, 29(4):1012–1024, 2008.

45 Jaroslav Nešetřil and Patrice Ossona de Mendez. First order properties on nowhere dense structures. *The Journal of Symbolic Logic*, 75(3):868–887, 2010.

46 Jaroslav Nešetřil and Patrice Ossona de Mendez. On nowhere dense graphs. *European Journal of Combinatorics*, 32(4):600–617, 2011.

47 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012.

48 Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409, 2001.

49 Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

50 Michael P. O'Brien and Blair D. Sullivan. Experimental evaluation of counting subgraph isomorphisms in classes of bounded expansion. *CoRR*, abs/1712.06690, 2017. `arXiv:1712.06690`.

51 Tobias Oelschlägel. *Treewidth from Treedepth*. Bachelor's thesis, RWTH Aachen University, Germany, 2014.

52 Michał Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. On the number of types in sparse graphs. *arXiv preprint arXiv:1705.09336*, 2017.

53 Felix Reidl. *Structural sparseness and complex networks*. Dr., Aachen, Techn. Hochsch., Aachen, 2016. Aachen, Techn. Hochsch., Diss., 2015. URL: `http://publications.rwth-aachen.de/record/565064`.

54 Felix Reidl, Fernando Sánchez Villaamil, and Konstantinos Stavropoulos. Characterising bounded expansion by neighbourhood complexity. *CoRR*, abs/1603.09532, 2016.

55 Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *International semantic Web conference*, pages 351–368. Springer, 2003.

56 Matei Ripeanu, Ian Foster, and Adriana Iamnitchi. Mapping the Gnutella Network. *IEEE Internet Computing*, 6(1):50–57, 2002.

57 Neil Robertson and Paul D. Seymour. Graph minors I–XXII, 1982–2010.

58 Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. URL: `http://networkrepository.com`.

59 Fernando Sanchez Villaamil. *About Treedepth and Related Notions.* Dissertation, RWTH Aachen University, Aachen, 2017. `doi:10.18154/RWTH-2017-09829`.

60 Sebastian Siebertz. Reconfiguration on nowhere dense graph classes. *CoRR*, abs/1707.06775, 2017.

61 Juliette Stehlé, N. Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, C. Régis, B. Lina, and P. Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8):e23176, 08 2011.

62 Jan van den Heuvel, Patrice Ossona de Mendez, Daniel A. Quiroz, Roman Rabinovich, and Sebastian Siebertz. On the generalised colouring numbers of graphs that exclude a fixed minor. *European Journal of Combinatorics*, 66:129–144, 2017.

63 Jan van den Heuvel, Stephan Kreutzer, Michał Pilipczuk, Daniel A. Quiroz, Roman Rabinovich, and Sebastian Siebertz. Model-checking for successor-invariant first-order formulas on graph classes of bounded expansion. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, pages 1–11, 2017.

64 Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440, 1998.

65 John G. White, Eileen Southgate, J. Nichol Thomson, and Sydney Brenner. The structure of the nervous system of the nematode caenorhabditis elegans: the mind of a worm. *Phil. Trans. R. Soc. Lond*, 314:1–340, 1986.

66 Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

67 Xuding Zhu. Colouring graphs with bounded generalized colouring number. *Discrete Math.*, 309:5562–5568, 2009.